

Amendments to the Specification:

Please amend the specification as follows:

Please replace paragraph bridging pages 16 and 17 (page 16, line 22 to page 17, line 6), with the following rewritten paragraph:

Moreover, the user instructs the user interface 103 to change the execution status. The user interface 103 interprets the instruction. When receiving an instruction to the changed execution status such as ~~temporal temporarily ceasing of~~ the debug object program during execution or resuming execution of the debug object program 120 in a temporary halt state, the ~~user~~ user interface 103 notifies the execution status ~~manager~~ manager of it. The execution-status manager 105 instructs the process manager 110 to reflect the status while changing its execution status.

Please replace the first full paragraph on page 30, lines 14-23, with the following rewritten paragraph:

The setting-status manager 104 ~~mangers~~ manages the common setting to all computers and the inherent setting to a specific computer. The common setting has an execution code of a debug object program and a BP table holding information about plural break points, and a variable table holding information about plural monitoring variables. The inherent setting has an inherent BP table holding a break point effective only inside the computer and an inherent variable table holding information about variable monitoring effective only inside the computer.

Please replace paragraph bridging pages 20 and 21 (page 20, line 24 to page 21, line 16), with the following rewritten paragraph:

The execution-status manager 105 ~~managers~~ manages the common setting to all computers and the inherent setting to a specific computer. The common setting has one of status variables --"in execution of a debugger, "in temporary halt", "in step execution", and "in non execution of program"--. The inherent setting has the process number of a process in execution of a debug object program. The communication object ~~manager~~ manager 107 has an information table having a combination of a network address of another computer at a communication destination and a port number for connecting with the communication section 106 at the address. The process manager 110 is connected to a debug object program in execution. The remote debugger activator 109 is connected to an external remote program executor 201. The external remote program executor 201 may be rsh of UNIX or the function of the debug object program executor, or a device made for the distributed debugger only.

Please replace paragraph bridging pages 23 and 24 (page 23, line 13 to page 24, line 6), with the following rewritten paragraph:

Fig. 5 is a flowchart representing the operation of the execution-status ~~manager~~ manager 105. The execution-status manager 105 receives an executions status change request in the step 501. The execution-status manager 105 first checks whether or not the change request corresponds to a change request for the same status as the current status in the step 502. If the status is the same as the current status, it is ignored. When the status differs from the current status, the change is reflected to the status in the step 503. Whether or not the request corresponds to an inherent status change is checked in the step 504. If the request is for an inherent status change, ~~noting~~ nothing is processed in the following steps. When the request is for a common status change, it is verified in the step 505 whether or not the process manager has sent the change request. When the change request is one sent from the process

manager, the changed content is issued to all other execution-status managers 105 connected via the communication section 106 in the step 507.

Please replace paragraph bridging pages 47 and 48 (page 47, line 16 to page 48, line 1), with the following rewritten paragraph:

If the distributed debugger system is realized without embodying the present invention, the user must manually change the status of other ~~buggers~~ debuggers when the execution status of part of debuggers which operates on plural computers changes, so that the serviceability to users degrades. When the user does not immediately change the status of other portions, the execution of other portions continues while part of distributed system to be debugged is temporarily halted, so that the internal status may be largely changed. This makes it very difficult for the user to grasp the execution status of a debug object program.